RECEIVED
CENTRAL FAX CENTER
MAR 26 2008

PROPOSED AMENDMENTS TO THE CLAIMS- US APP 10/674,223

Please consider the claims as follows.

1. (Currently Amended) A method for providing verification for a simulation design, comprising:

obtaining the simulation design comprising a programming language interface system call;

~~encoding a target of~~ obtaining a reference value corresponding to the programming language interface system call;

encoding the reference value into the simulation design to obtain a ~~first~~ modified simulation design;

~~modifying the programming language interface system call to reference the target in the first modified simulation design to obtain a second modified simulation design; and~~

compiling the modified simulation design to obtain a compiled simulation design; and

verifying the ~~second modified~~ compiled simulation design using a simulation testbench.

2. (Original) The method of claim 1, wherein the simulation design comprises a register transfer level design.

3. (Currently Amended) The method of claim 1, wherein encoding the ~~target~~ reference value of the programming language interface system call comprises:

obtaining a set of hardware state elements from the simulation design;

obtaining a set of high-level state elements from a high-level design;

determining a common set of state elements from the set of hardware state elements and the set of high-level state elements;

identifying at least one relationship between the high-level state element and the hardware state element in the common set of state elements;

determining whether the at least one relationship is influenced by a test vector to obtain an influenced set of relationships; and

encoding the influenced set of relationships to obtain [[a]] the modified simulation design.

4. (Currently Amended) The method of claim 3, wherein encoding the influenced set of relationships comprises:

obtaining high-level state element values for the high-level state elements in the influenced set of relationships from a simulation of the high-level design; and

storing the high-level state element values in an array, wherein at least one of the high-level state element values is the reference value.

5. (Cancel)

6. (Currently Amended) The method of claim 3, wherein the ~~hardware~~ high-level state element comprises at least one from the group consisting of a processor register and a memory.

7. (Currently Amended) The method of claim 3, wherein the ~~high-level~~ hardware level state element comprises at least one from the group consisting of a flip-flop, a latch, and a memory.

8. (Original) The method of claim 3, wherein the relationship comprises at least one of the group consisting of a one-to-one relationship, a one-to-many relationship, and a relationship defined by a mathematical function.

9. (Currently Amended) The method of claim 1, wherein the ~~second modified hardware level~~ compiled simulation design is verified on a hardware-based simulation test bench.

10. (Currently Amended) A computer system for providing verification for a simulation design, comprising:

a processor;

a memory;

a storage device; and

software instructions stored in the memory for enabling the computer system to perform:

obtaining the simulation design comprising a programming language interface system call;

~~encoding a target of~~ obtaining a reference value corresponding to the programming language interface system call;

encoding the reference value into the simulation design to obtain a ~~first~~ modified simulation design;

~~modifying the programming language interface system call to reference the target in the first modified simulation design to obtain a second modified simulation design; and~~

compiling the modified simulation design to obtain a compiled simulation design; and

verifying the ~~second modified~~ compiled simulation design using a simulation testbench.

11. (Currently Amended) The computer system of claim 10, wherein the simulation design comprises a ~~resister~~ register transfer level design.

12. (Currently Amended) The computer system of claim, 10, wherein encoding the ~~target~~ reference value of the programming language interface system call comprises:

obtaining a set of hardware state elements from the simulation design;

obtaining a set of high-level state elements from a high-level design;

determining a common set of state elements from the set of hardware state elements and the set of high-level state elements;

identifying at least one relationship between the high-level state element and the hardware state element in the common set of state elements;

determining whether the at least one relationship is influenced by the test vector to obtain an influenced set of relationships; and

encoding the influenced set of relationships to obtain [[a]] the modified simulation design.

13. (Currently Amended) The computer system of claim 12, wherein encoding the influenced set of relationships comprises:

obtaining high-level state element values for the high-level state elements in the influenced set of relationships from a simulation of the high-level design; and

storing the high-level state element values in an array, wherein at least one of the high-level state element values is the reference value.

14. (Cancel)

15. (Currently Amended) The computer system of claim 12, wherein the ~~hardware~~ high-level state element comprises at least one from the group consisting of a processor register and a memory.

16. (Currently Amended) The computer system of claim 12, wherein the ~~high-level~~ hardware state element comprises at least one from the group consisting of a flip-flop, a latch, and a memory.

17. (Original) The computer system of claim 12, wherein the relationship comprises at least one of the group consisting of a one-to-one relationship, a one-to-many relationship and a relationship defined by a mathematical function.

18. (Currently Amended) The computer system of claim 10, wherein the ~~second modified hardware-level~~ modified simulation design is verified on a hardware-based simulation test bench.

19. (Cancel)

20. (Currently Amended) Apparatus providing verification for a simulation design, comprising:

means for obtaining the simulation design comprising a programming language interface system call;

means for ~~encoding a target of~~ obtaining a reference value corresponding to the programming language interface system call;

means for encoding the reference value into the simulation design to obtain a ~~first~~ modified simulation design;

~~means for modifying the programming language interface system call to reference the target in the first modified simulation design to obtain a second modified simulation design; and~~

means for compiling the modified simulation design to obtain a compiled simulation design; and

means for verifying the ~~second modified~~ compiled simulation design using a simulation testbench.

21. (New) The apparatus of claim 20, wherein the simulation design comprises a register transfer level design.

22. (New) The apparatus of claim 20, wherein encoding the reference value of the programming language interface system call comprises:

means for obtaining a set of hardware state elements from the simulation design;

means for obtaining a set of high-level state elements from a high-level design;

means for determining a common set of state elements from the set of hardware state elements and the set of high-level state elements;

means for identifying at least one relationship between the high-level state element and the hardware state element in the common set of state elements;

means for determining whether the at least one relationship is influenced by a test vector to obtain an influenced set of relationships; and

means for encoding the influenced set of relationships to obtain the modified simulation design.

23. (New) The apparatus of claim 22, wherein encoding the influenced set of relationships comprises:

means for obtaining high-level state element values for the high-level state elements in the influenced set of relationships from a simulation of the high-level design; and

means for storing the high-level state element values in an array, wherein at least one of the high-level state element values is the reference value.